CSCE 2301- Digital Design 1

Project 2 Report – Fall-2024

Pong Game

John Ebrahim

Ziad Gaballah

Ahmed Bamadhaf

Supervised by

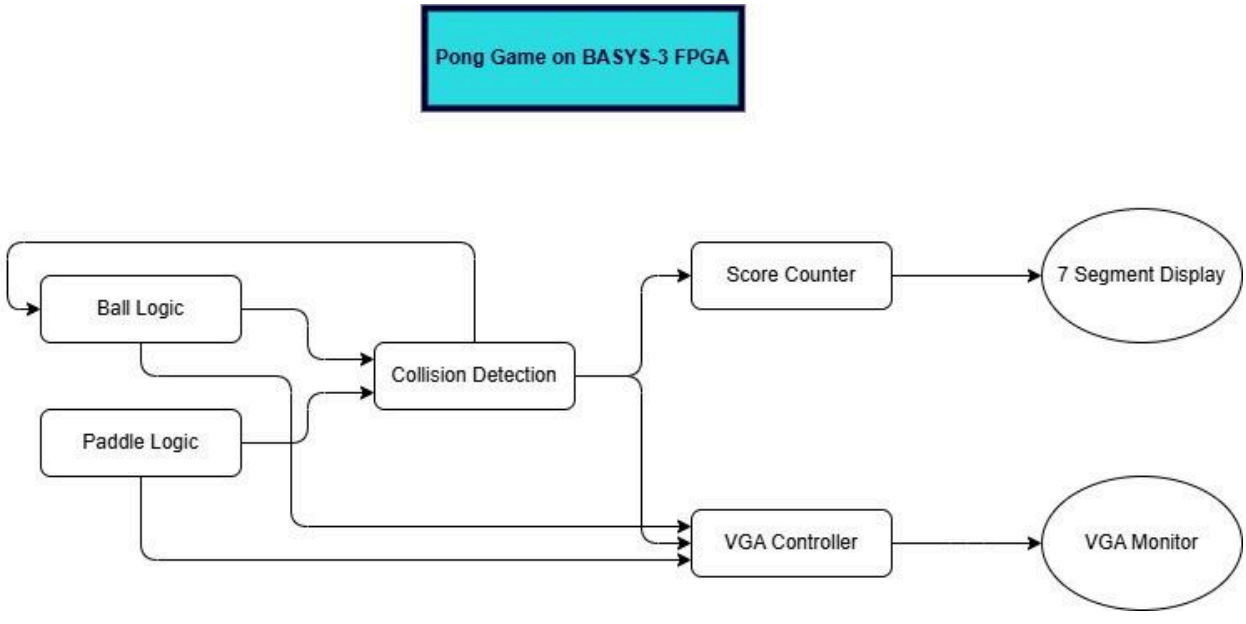
Dr. Dina Mahmoud

This project brings the well-known game Pong to life on the Basys 3 FPGA board using Verilog, a hardware description language. The goal is to recreate this game while learning the basics of digital design and gaining experience with FPGA programming. Pong is a two-player game where each player controls a paddle to hit a ball back and forth. Points are scored when one player fails to hit the ball back to the other player.

The project is broken into key components:

1. Paddle Control: Players use the buttons or switches on the FPGA board to move their paddles up and down.
2. Ball Movement: The ball moves dynamically across the screen, bouncing off paddles and screen edges.
3. Collision Detection: Ensures the ball interacts properly with paddles and boundaries, updating its direction and triggering a score if missed.
4. Scorekeeping: Tracks the score for both players and displays it on the FPGA's 7-segment display on the FBGA.
5. VGA Display: The game is visually presented on a monitor, showing the paddles, ball, and scores.

Milestone 1:



Block diagram

For the first milestone, we have created a this block diagram that basically contains the following:

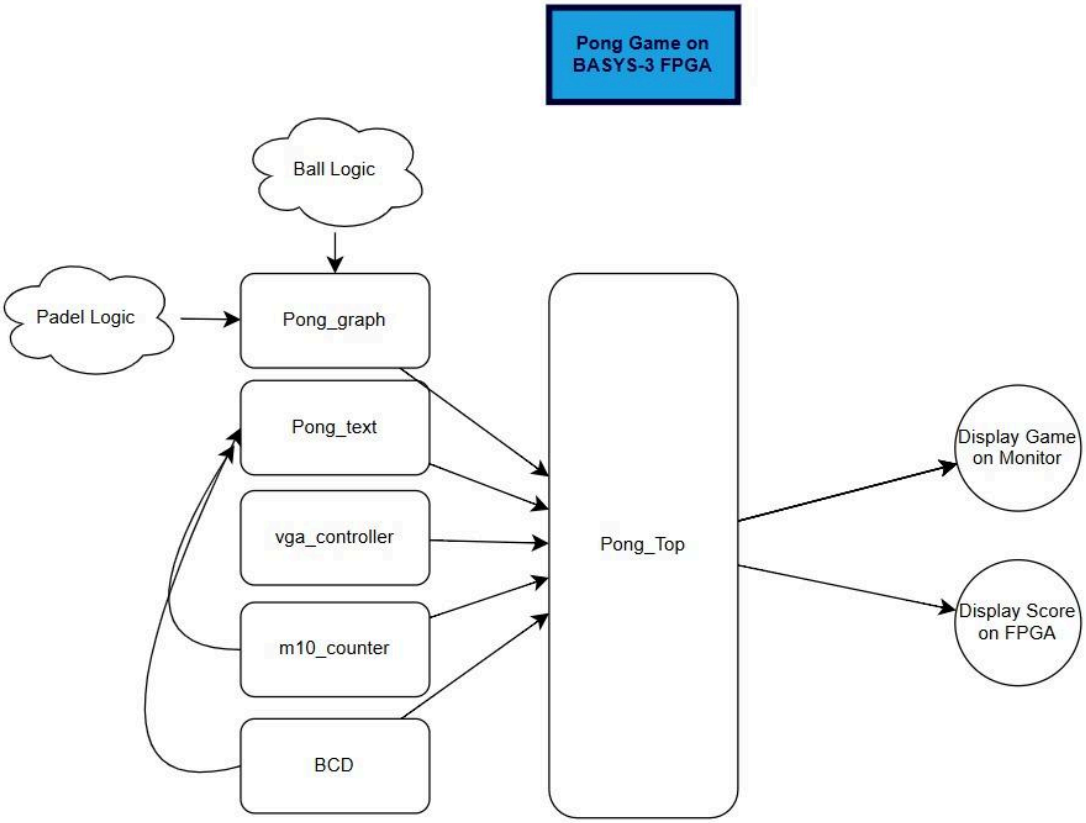
1. Ball Logic:
    - This part controls how the ball moves on the screen. It decides the ball's speed and direction and reacts when it hits something (like a paddle or a wall).
    - It shares the ball's position with the Collision Detection block to check for hits.
2. Paddle Logic:
    - This handles the movement of the paddles based on player input. Players use the buttons or switches on the FPGA board to move their paddles up or down.
    - The paddle positions are sent to the Collision Detection block to check if the ball hits them.
3. Collision Detection:
    - This block makes sure the ball behaves correctly when it collides with paddles or walls. If the ball hits a paddle, it bounces back; if it misses, a point is scored for the other player.
    - It communicates with the Ball Logic to adjust the ball's movement and with the Score Counter to update the score.
4. Score Counter:
    - Keeps track of each player's score and sends it to the 7-Segment Display so players can see the current scores in real time.
5. VGA Controller:
    - This creates the game visuals on a monitor. It displays the paddles, ball, and scores by using inputs from the Ball Logic, Paddle Logic, and Collision Detection blocks.
6. VGA Monitor:
    - The VGA monitor is where players can see the game happening, including the paddles, ball, and scores.
7. 7-Segment Display:
    - The scores are also displayed on the FPGA board's built-in 7-segment displays, giving players a quick view of the results.

However, since the first milestone required us to have a displayed monitor with colors on it, we did not implement any of these blocks into the code.

We created a module that can color the monitor in different colors using the VGA. We have designed a module that basically has three colors (RGB) and each color has four bits. Each bit of those colors was connected to a switch on the FBGA. This allowed us to mix colors on the monitor when switching different bits in different colors.

Unfortunately, during this milestone, we could not display pixels with different colors on the screen.

Second/Third milestone :



Explanation of the Block Diagram

This block diagram represents the design of the Pong Game implemented on the Basys-3 FPGA board. Each module has a distinct role in creating the game, and they work together to manage gameplay, visuals, and scoring. Below is a detailed explanation of each module, how they interact, and the features added to the game:

1. Ball Logic

- The Ball Logic module determines the ball's movement across the screen, including speed, direction, and bouncing behavior.
- It updates the ball's position and communicates with Pong_graph for display and collision detection. If the ball hits a paddle or screen edge, it adjusts direction accordingly.
- If the ball reaches the right or left edge of the screen, it signals a score update and decrements the ball counter if applicable.

2. Paddle Logic

- This module handles player input from buttons or switches on the FPGA board to control paddle movement (up and down).
- It ensures paddles stay within the boundaries of the playing field and sends their positions to Pong_graph for collision detection and graphical updates.

3. Pong_graph

- Pong_graph is the main visualmodule, responsible for rendering game elements, including the ball, paddles, and boundaries, on the screen.
- It interacts with Ball Logic and Paddle Logic to detect collisions between the ball and paddles. When a collision occurs, it informs Ball Logic to change the ball's direction.
- Pong_graph also triggers score updates when the ball passes a paddle (hits the edge of the screen).

4. Pong_text

- Pong_text is used to display textual elements in the game, including:

    - "Score": This label appears on the monitor to indicate where player scores are displayed.
    - Scores: If the ball hits the right edge, the left player's score increments by one, and vice versa.
    - Ball Counter: Tracks and displays the number of balls remaining in the game.
- This module ensures the text is updated in sync with gameplay and is positioned correctly on the monitor.

5. VGA Controller

- The VGA Controller generates the signals needed to display the game on a monitor.
- It combines the graphical output from Pong_graph and the text output from Pong_text to create the final visual display.

6. m10_counter

- This module provides timing for the game. It controls:
    - The one-second display duration for the "Game Over" message.
    - The ball's speed and paddle response timing to ensure smooth gameplay.
    - The synchronization of the ball counter updates.

7. BCD (Binary Coded Decimal)

- This module converts binary score values into a format suitable for display on the FPGA's 7-segment displays.
- We implemented this BCD converter during a previous lab exercise and adapted it for use in this project.
- It receives updated scores from Pong_graph and ensures they are displayed in real-time on the FPGA board.

8. Pong_Top

- Pong_Top is the top-level module that integrates all the components. It ensures proper communication and coordination between Ball Logic, Paddle Logic, Pong_graph, Pong_text, VGA Controller, m10_counter, and BCD.

Bouns features

1. **Score Display Logic:**
    - When the ball hits the right edge of the screen, the left player's score increments by one. Similarly, if the ball hits the left edge, the right player's score increments.
    - Scores are displayed both on the monitor (via Pong_text) and the FPGA board's 7-segment display (via BCD).
2. **Ball Counter:**
    - Tracks the number of balls left in the game. The ball counter decreases by one each time a point is scored.
    - When the counter reaches zero, the game ends, and a "Game Over" message is displayed.
3. **Game Over Message:**
    - Once a player wins, a "Game Over" message appears on the monitor for exactly one second. The game then resets, allowing for a new round to begin.
4. **Ball Speed Control:**
    - A switch on the FPGA board allows players to change the ball's speed dynamically during the game.

The interactions :

1. **Player Input and Paddle Movement:**
    - Players use buttons or switches to control paddle movement. Paddle Logic sends updated positions to Pong_graph for display and collision detection.
2. **Ball Movement and Collision Detection:**
    - Ball Logic calculates the ball's position and interacts with Pong_graph for collision detection. When a collision is detected, Ball Logic adjusts the ball's trajectory accordingly.

3. **Score Updates and Ball Counter:**

    - Pong_graph detects when the ball reaches the edge of the screen and triggers a score update. It also signals m10_counter to decrement the ball counter.

4. **Display Output:**

    - Pong_graph and Pong_text send graphical and textual data to the VGA Controller, which combines them to display the game on a monitor.
    - Scores and the ball counter are updated on the 7-segment display using the BCD module.

Challenges :

- At first, we faced problems trying to cover the entire screen with our output but we managed to fix it later on
- Another challenge was the control logic of the second paddle as it was moving simultaneously with the first paddle and it was not independent but it was fixed
- Similarly, we faced issues trying to isolate the score of both players. This was fixed but cost us a lot of time
- A limitation to our design is the if both players try to move the paddles at the same time
- We were planning to implement the movement using a keyboard connected with FBGA, however due to time constraints we could not